

**Amendments to the Claims:**

Please amend the claims as follows:

1. (Currently Amended) A concurrent code load apparatus for fastload code image update on a communications adapter, the apparatus comprising:
  - an image load module configured to load a copy of a new code image in a memory on the communications adapter, the memory concurrently storing a copy of an old code image used by the communications adapter;
  - a memory initialization module configured to invoke the new code image to perform a memory initialization operation while the old code image is still executing; and
  - an image overlay module configured to overlay the old code image with the new code image;
  - a query module configured to determine an incompatibility between the old code image and the new code image; and
  - an image bridge module configured to reconcile the incompatibility between the old code image and the new code image while the old code image is still executing.
2. (Cancelled)
3. (Cancelled)
4. (Original) The apparatus of claim 1, further comprising a fastload key module configured to create and store a fastload key to indicate a fastload code image update on the communications adapter.

5. (Original) The apparatus of claim 1, further comprising a fastload adapter initialization module configured to initialize the communications adapter using a fastload initialization sequence in response to a fastload code image update.

6. (Original) The apparatus of claim 5, wherein the fastload adapter initialization module is further configured to access a fastload key prior to using the fastload initialization sequence.

7. (Original) The apparatus of claim 1, further comprising a standard adapter initialization module configured to initialize the communications adapter using a standard initialization sequence in response to a failure to access a fastload key.

8. (Original) The apparatus of claim 7, further comprising a fastload adapter initialization module configured to execute a fastload initialization operation during the standard initialization sequence.

9. (Original) The apparatus of claim 1, wherein the old code image comprises a code image update module configured to control a code image update.

10. (Original) The apparatus of claim 9, wherein the code image update module comprises a load module configured to load the new code image in the memory.

11. (Original) The apparatus of claim 9, wherein the code image update module comprises a branch module configured to branch from the old code image to the new code image.

12. (Original) The apparatus of claim 1, wherein the new code image comprises a bootstrap module configured to define a bootstrap operation, the bootstrap operation configured to facilitate a code image update.

13. (Currently Amended) The apparatus of claim 1, wherein the bootstrap module comprises ~~the image bridge module a conversion module, the image bridge module configured to reconcile an incompatibility between the old code image and the new code image using the conversion module.~~

14. (Currently Amended) The apparatus of claim 1, wherein the bootstrap module comprises ~~the image overlay module a copy module~~, the image overlay module configured to overlay the old code image with the new code image using the copy module.

15. (Currently Amended) A storage system for facilitating fastload code image update on a source communications adapter, the storage system comprising:

    a source input device configured to receive a source electronic storage media device, the source electronic storage media device configured to store a new code image;

    a storage system processor configured to initiate ~~the a~~ fastload code image update and notify the source communications adapter of the fastload code image update; ~~and~~

    the source communications adapter configured to copy the new code image to a local memory device and to implement the fastload code image update; ~~and,~~

the storage system processor further configured to determine and reconcile an incompatibility between the old code image and the new code image while the old code image is still executing.

16. (Currently Amended) A method for fastload code image update on a communications adapter, the method comprising:

loading a copy of a new code image in a memory on the communications adapter, the memory concurrently storing a copy of an old code image used by the communications adapter;

invoking the new code image to perform a memory initialization operation while the old code image is still executing; and

overlaid the old code image with the new code image; determining an incompatibility between the old code image and the new code image; and

reconciling the incompatibility between the old code image and the new code image while the old code image is still executing.

17. (Cancelled)

18. (Cancelled)

19. (Original) The method of claim 16, further comprising creating and storing a fastload key to indicate a fastload code image update on the communications adapter.

20. (Original) The method of claim 16, further comprising initializing the communications adapter using a fastload initialization sequence in response to a fastload code image update.

21. (Original) The method of claim 20, further comprising determining if access a fastload key prior to using the fastload initialization sequence.

22. (Original) The method of claim 16, further comprising initializing the communications adapter using a standard initialization sequence in response to a failure to access a fastload key.

23. (Original) A method for fastload code image update on a communications adapter, the method comprising:

loading a copy of a new code image in a memory on the communications adapter, the memory concurrently storing a copy of an old code image used by the communications adapter;

invoking the new code image to perform a memory initialization operation while the old code image is still executing;

~~identifying a characteristic of the old code image;~~

determining identifying an incompatibility between the old code image and the new code image;

reconciling the incompatibility between the old code image and the new code image while the old code image is still executing;

overlaying the old code image with the new code image;

creating and storing a fastload key to indicate a fastload code image update on the communications adapter; and

initializing the communications adapter using a fastload initialization sequence in response to a fastload code image update.

24. (Currently Amended) A computer readable storage medium comprising computer readable code configured to carry out a method for fastload code image update on a communications adapter, the method comprising:

loading a copy of a new code image in a memory on the communications adapter, the memory concurrently storing a copy of an old code image used by the communications adapter;

invoking the new code image to perform a memory initialization operation while the old code image is still executing; and

overlaying the old code image with the new code image;  
determining an incompatibility between the old code image and the new code image; and

reconciling the incompatibility between the old code image and the new code image while the old code image is still executing.

25. (Cancelled)

26. (Cancelled)

27. (Original) The computer readable storage medium of claim 24, wherein the method further comprises creating and storing a fastload key to indicate a fastload code image update on the communications adapter.

28. (Original) The computer readable storage medium of claim 24, wherein the method further comprises initializing the communications adapter using a fastload initialization sequence in response to a fastload code image update.

29. (Original) The computer readable storage medium of claim 28, wherein the method further comprises determining if access a fastload key prior to using the fastload initialization sequence.

30. (Original) The computer readable storage medium of claim 24, wherein the method further comprises initializing the communications adapter using a standard initialization sequence in response to a failure to access a fastload key.

31. (Original) The computer readable storage medium of claim 24, wherein the old code image comprises a code image update module configured to control a code image update.

32. (Original) The computer readable storage medium of claim 31, wherein the code image update module comprises a load module configured to load the new code image in the memory.

33. (Original) The computer readable storage medium of claim 31, wherein the code image update module comprises a branch module configured to branch from the old code image to the new code image.

34. (Original) The computer readable storage medium of claim 24, wherein the new code image comprises a bootstrap module configured to define a bootstrap operation, the bootstrap operation configured to facilitate a code image update.

35. (Original) The computer readable storage medium of claim 34, wherein the bootstrap module comprises a conversion module, the image bridge module configured to reconcile an incompatibility between the old code image and the new code image using the conversion module.

36. (Original) The computer readable storage medium of claim 34, wherein the bootstrap module comprises a copy module, the image overlay module configured to overlay the old code image with the new code image using the copy module.

37. (Currently Amended) An apparatus for fastload code image update on a communications adapter, the apparatus comprising:

means for loading a copy of a new code image in a memory on the communications adapter, the memory concurrently storing a copy of an old code image used by the communications adapter;

means for invoking the new code image to perform a memory initialization operation while the old code image is still executing; and

means for overlaying the old code image with the new code image;

means for determining an incompatibility between the old code image and the new code image; and

means for reconciling the incompatibility between the old code image and the new code image while the old code image is still executing.